



Efficient solution of many instances of a simulation-based optimization problem utilizing a partition of the decision space

Downloaded from: <https://research.chalmers.se>, 2023-05-04 20:23 UTC

Citation for the original published paper (version of record):

Nedelkova, Z., Lindroth, P., Patriksson, M. et al (2018). Efficient solution of many instances of a simulation-based optimization problem utilizing a partition of the decision space. *Annals of Operations Research*, 265(1): 93-118.
<http://dx.doi.org/10.1007/s10479-017-2721-y>

N.B. When citing this work, cite the original published paper.

Efficient solution of many instances of a simulation-based optimization problem utilizing a partition of the decision space

Zuzana Nedělková¹  · Peter Lindroth² ·
Michael Patriksson¹ · Ann-Brith Strömberg¹

Published online: 8 December 2017

© The Author(s) 2017. This article is an open access publication

Abstract This paper concerns the solution of a class of mathematical optimization problems with simulation-based objective functions. The decision variables are partitioned into two groups, referred to as variables and parameters, respectively, such that the objective function value is influenced more by the variables than by the parameters. We aim to solve this optimization problem for a large number of parameter settings in a computationally efficient way. The algorithm developed uses surrogate models of the objective function for a selection of parameter settings, for each of which it computes an approximately optimal solution over the domain of the variables. Then, approximate optimal solutions for other parameter settings are computed through a weighting of the surrogate models without requiring additional expensive function evaluations. We have tested the algorithm's performance on a set of global optimization problems differing with respect to both mathematical properties and numbers of variables and parameters. Our results show that it outperforms a standard and often applied approach based on a surrogate model of the objective function over the complete space of variables and parameters.

Keywords Simulation-based optimization · Surrogate model · Response surface · Partition of variables · Tyres

✉ Zuzana Nedělková
zuzana@chalmers.se

Peter Lindroth
peter.lindroth@volvo.com

Michael Patriksson
mipat@chalmers.se

Ann-Brith Strömberg
anstr@chalmers.se

¹ Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, 412 96 Gothenburg, Sweden

² Chassis and Vehicle Dynamics Engineering, Global Operations, Volvo Group Trucks Technology, 405 08 Gothenburg, Sweden

Mathematics Subject Classification 90C27 · 90C56 · 90B06

1 Introduction

This article proposes a new algorithm that utilizes a *response surface* method (see Jones 2001 for a review of such methods) for solving many similar instances of a computationally expensive simulation-based optimization problem.

The research leading to this paper is motivated by a project conducted by Volvo GTT focusing on the optimization of truck tyre design (Lindroth 2015; Šabartová et al. 2014). The purpose is to enable—for each combination of truck configuration and operating environment—the identification of a tyre design that minimizes the energy losses caused by the tyres. Hence, we aim to solve a large set of instances of a simulation-based optimization problem, where the vehicle configuration and operating environment vary among the instances. No attempt to solve such a set of problem instances—other than solving each instance separately—has been found in the literature. The suggested methodology is efficient enough to enable the solution of the computationally expensive truck tyre design problem for each individual customer during the sales process.

First, our methodology uses a standard algorithm for simulation-based optimization (Gutmann 2001) to compute optimal tyre designs, described by continuous variables (e.g., tyre diameter and inflation pressure). This is performed for each of a selection of so-called *strategic vehicle specifications* (SVSs; see Lindroth 2011, Section 2), described by discrete-valued parameters. These optimal tyre designs—as well as a priori information about the properties of the simulation-based functions—are then utilized for an efficient computation of approximately optimal tyre designs for many other vehicle specifications. This methodology is intended to be incorporated into Volvo GTT's existing sales tool. Utilizing a variety of artificial and real test problems, we compare the accuracy of the approximately optimal solution for a non-strategic parameter setting with that achieved by other approaches. The first approach is to construct a surrogate model that is valid for all possible parameter settings and the second one is to use a standard algorithm for solving simulation-based optimization problems for each parameter setting (both strategic and non-strategic). It was identified that the methodology proposed yields more accurate solutions than other approaches for most of the test problems considered (Sect. 4 provides details).

We assume that the variation of the values of the computationally expensive functions is larger with respect to the *variables* than with respect to the *parameters*; an alternative assumption is that the parameters' principal influence on the function values is known a priori. As for the tyre design application, the value of the energy loss function is influenced more significantly by the tyre design (i.e., the variables) than by the vehicle configuration and the operating environment (i.e., the parameters).

Many algorithms for solving simulation-based optimization problems utilize a simple surrogate model of the computationally expensive function (Conn et al. 2009, Chs. 10–12). Our suggested methodology is initiated by constructing and optimizing a surrogate model of the simulation-based function for each of a selection of parameter settings (here also referred to as problem instances). Then, for other parameter settings, approximate surrogate models of the simulation-based function are assembled (using no additional simulations) and minimized, resulting in an approximate optimal solution for each parameter setting; see also Fig. 2.

The algorithm presented in this paper can—besides the tyre design problem—be used in other applications in which a number of similar computationally expensive optimization problem instances are to be solved. Examples include the design of freight aircraft to be used for several types of transport missions (Willcox and Wakayama 2003), and the optimization of the charge of melting with respect to the quality of various products and which needs to be done in real time (Dupačová and Popela 2005).

1.1 Previous work

Increasingly complex engineering optimization problems are modeled and solved, due to the increased possibility to use numerical simulations (such as, e.g., finite elements methods for structural analysis and computational fluid dynamics). The simulation-based objective function of such an optimization problem is often treated as a black box, due to the lack of an explicit expression of the function. As a consequence, no favourable properties, such as convexity or differentiability, of the function can be inferred. These types of problems—classified as simulation-based optimization problems (Fu 2014)—can in practice not be solved by algorithms requiring many function evaluations, such as direct search methods (Lewis et al. 2000). Local optimization algorithms, such as the MADS algorithm (Audet and Dennis 2006), find local optima of the function. However, many optimization problems of practical relevance are non-convex and exhibit multiple local optima, thus demanding the use of global optimization techniques for their solution. A common methodology for finding an approximate global optimum is to employ a *response surface method* (RSM; see Jones 2001). An RSM provides a response function that mimics the behaviour of the computationally expensive objective function as closely as possible, while being computationally tractable; the response function is then optimized. *Radial basis function* (RBF) interpolation (Wendland 2005) and *Kriging interpolation* (Lophaven et al. 2002) are widely used to model multivariate functions and often yield good global representations of the unknown functions; see Buhmann (2003). Local and global optimization methods based on RBF interpolation can be studied in Wild et al. (2008) and Gutmann (2001), respectively.

The idea of partitioning the decision variables into two groups, here denoted variables and parameters, can be found in the area of bilevel optimization (Colson et al. 2007) where the objective function is optimized over both groups of variables simultaneously. Here we assume that the values of the parameters are set before the objective function is optimized over the resulting feasible region for the variables. For many applications the influence of the parameters/variables on the objective function is indicated in the documentation about the model used (e.g., in the form of validations with respect to various inputs) or it can be measured by parameter studies or sensitivity analyses. We will utilize this difference in the degree of influence to reduce the dimension of the search space; see Sect. 3. The idea of reducing the search space, guided by additional information about the decision variables, is utilized also within constraint programming (Bessiere 2006; Stuber et al. 2015).

While RSMs are typically utilized to accurately predict the response function over the entire feasible region, we first compute an accurate prediction of the response function over the variables' domain for each of the selected parameter settings. For other settings of the parameter values the response function is then defined by a weighted sum of the interpolation coefficients that define the predicted response functions; our approach to determine the weights is described in Sect. 3.

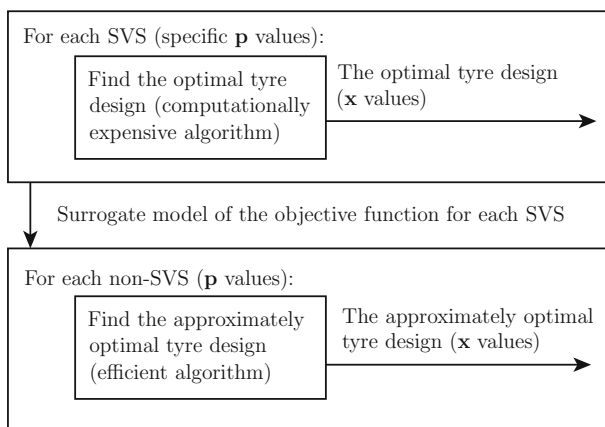


Fig. 1 Illustration of the proposed algorithm for the truck tyre design problem. Both the operating environment specification and the vehicle configuration are represented by parameters, \mathbf{p} , while the tyre design is represented by the decision variables, \mathbf{x} . Optimal solutions to the design problem for the SVSs operating in the most common environments are found using a computationally expensive optimization method, after which approximately optimal solutions for other vehicle configurations/operating environment specifications are computed efficiently by the proposed methodology

1.2 Motivation

The present sales tool at Volvo GTT generates a large set of feasible tyres for each truck. The truck tyres selection process is then based on experience and customer input that can be further improved by means of scientific methodologies. Volvo GTT's ultimate goal is to find an optimal design of the tyres for each vehicle configuration and operating environment specification, with respect to minimizing the energy losses caused by the tyres. In the present phase of our research project (Lindroth 2015), the tyres are described by continuous design variables (tyre diameter, tyre width, and inflation pressure). Our future research will introduce discrete design variables representing the tyres available in a tyre database, i.e., solving the so-called *tyres selection problem*.

The number of vehicle configurations that can be manufactured by Volvo is huge; the same holds for the variety of environments in which the vehicles are operated. Solving the tyre design problem is very time consuming and computationally expensive, since (1) for each operating environment a sufficiently detailed vehicle model—including finite element models of the tyres—has to be run, and (2) the vehicle specification and operating environment models have to be adapted to each customer. One can afford to search for an optimal tyre design only for a small subset of all combinations of vehicle configuration and environment specification, i.e., only for the SVSs operating in the most common environments. For the non-SVS vehicles, the tyre design problem must be solved in real time during the sales process. Hence, for the non-SVS vehicles an algorithm that quickly finds an approximately optimal tyre design is needed in place of the computationally intractable search for an optimal tyre design.

Most of the energy losses caused by the tyres stem from the rolling resistance (Walter and Conant 1974). The main ingredient in our model of the tyre design problem is therefore the rolling resistance coefficient (RRC). A finite element model of a truck tyre (Ali et al. 2013) was used; this model allows for an investigation of the influence on the RRC of the tyre design variables and of the parameters describing the vehicle and the operating environment.

Each evaluation of the model takes, however, around four hours of computation time. The function that describes the RRC is influenced more significantly by the values of the tyre design variables than with those of the parameters describing the vehicle and its typical operating environment; see Ali et al. (2013). This property of the RRC will be used for assembling designs for non-SVSs vehicles, based on their corresponding positions in the parameter space; see Fig. 1.

1.3 Outline

The remainder of this article is organized as follows. Algorithms utilized for the minimization of simulation-based functions, with a main focus on RSMs based on RBF interpolation, are reviewed in Sect. 2. The algorithm suggested for solving many instances of a simulation-based optimization problem with a partitioned decision space is presented in Sect. 3. The methodology used to assess the performance of the algorithm, the set of test problems selected, and the results from the test are presented in Sect. 4. Finally, Sect. 5 provides conclusions as well as topics for future research.

2 Minimizing simulation-based functions

We aim to solve a set of simulation-based optimization problems possessing computationally expensive black-box objective functions. Let $\mathbf{x} \in \mathbb{R}^M$ denote the vector of decision variables, $\mathbf{p} \in \mathbb{R}^{D-M}$ with $1 \leq M \leq D^1$ denotes the vector of parameters, and $f : \mathbb{R}^D \mapsto \mathbb{R}$ is the computationally expensive objective function, which is evaluated through simulations. The vectors $\mathbf{l}_x < \mathbf{u}_x \in \mathbb{R}^M$ and $\mathbf{l}_p < \mathbf{u}_p \in \mathbb{R}^{D-M}$ define lower and upper bounds on the variables and parameters, respectively. The simulation-based optimization problem considered is then to

$$\underset{\mathbf{x} \in [\mathbf{l}_x, \mathbf{u}_x]}{\text{minimize}} \quad f(\mathbf{x}, \mathbf{p}), \quad (1)$$

for possibly many values of $\mathbf{p} \in [\mathbf{l}_p, \mathbf{u}_p]$. The requirement that $\mathbf{x} \in [\mathbf{l}_x, \mathbf{u}_x]$ is referred to as *box constraints*.

In this section we examine methods for the optimization of the problem (1) over \mathbf{x} . We assume that the function modelling the RRC [represented in (1) by the function f] is continuous, which motivates the use of *continuous surrogate models*; cf. Sect. 2.1. Since the simulations of the objective function are computationally expensive the number of function evaluations should be kept at a minimum. Within the engineering community, RSMs are popular for solving such simulation-based optimization problems (Jones 2001).

2.1 Response surface methods (RSMs)

Response surfaces provide fast computations of surrogate functions in place of time-consuming simulations. By running the simulation for a set of sample points and fitting an analytical function to the resulting function values, a computationally cheap surrogate model of the function is obtained and can be used for optimization purposes. The initial set of sample points may, e.g., be determined by a *design of experiments* technique (see the review in Simpson et al. 2001). Then additional points are sampled and evaluated (through a simulation) in order to refine the approximation of the true function, until a stopping criterion is met (e.g., a maximum allowed number of simulations is attained, or the function value

¹ $\mathbf{p} \in \mathbb{Z}^{D-M}$ can also be considered with the same methodology.

being below a given target). The strategies for selecting the sample points differ among specific algorithms. An efficient strategy should balance local and global searches (Jakobsson et al. 2010), such that the information from the surrogate model is utilized and no interesting part of the variable domain is left unexplored. Algorithm 1 summarizes a general RSM.

Algorithm 1 General response surface method (RSM) for the optimization problem (1) for $\mathbf{p} = \bar{\mathbf{p}}$

- 0: Create an initial set of sample points $\bar{\mathbf{x}}^n \in [\mathbf{l}_x, \mathbf{u}_x]$, $n \in \{1, \dots, N\}$, and evaluate $f(\mathbf{x}, \bar{\mathbf{p}})$ for all $\mathbf{x} \in \{\bar{\mathbf{x}}^1, \dots, \bar{\mathbf{x}}^N\}$.
 - 1: Construct a surrogate model of $f(\cdot, \bar{\mathbf{p}})$ from the points evaluated, i.e., $(\bar{\mathbf{x}}^n, f(\bar{\mathbf{x}}^n, \bar{\mathbf{p}}))$, $n \in \{1, \dots, N\}$.
 - 2: Select and evaluate a new sample point $\bar{\mathbf{x}}^{N+1}$ to refine the surrogate model.
 - 3: If a stopping criterion is met, exit. Otherwise, let $N := N + 1$ and go to step 1.
-

Assuming that the simulations of the true function do not exhibit a high degree of computational noise, the shape of the function is usually better captured by an interpolating response surface—passing through all sample points—than by a non-interpolating surface—found by, e.g., minimizing the sum of squared errors at the sample points—which may be quite distant from the function values at the sample points (Wendland 2005, Section 1.5). The interpolating surface is typically constructed as a linear combination of *polynomial* and *basis function* terms. Our development is based on an RSM which uses an RBF interpolation, originally presented in Powell (1999). Our algorithm can, however, utilize any kind of response surface (here also referred to as surrogate model).

2.2 Radial basis function interpolation

The surrogate model based on an RBF interpolation is formally defined below.

Definition 1 (*radial basis function, RBF*) Letting $\|\cdot\|$ denote the Euclidean norm, a function $g : \mathbb{R}^M \mapsto \mathbb{R}$ is called a *radial basis function* if there exists a univariate function $\phi : [0, \infty) \mapsto \mathbb{R}$ such that

$$g(\mathbf{x}) = \phi(\|\mathbf{x}\|), \quad \mathbf{x} \in \mathbb{R}^M.$$

Assume that the objective function $f(\cdot, \bar{\mathbf{p}})$ of (1) is simulated at N distinct sample points $\bar{\mathbf{x}}^n \in [\mathbf{l}_x, \mathbf{u}_x] \subset \mathbb{R}^M$, $n = 1, \dots, N$. Denote the vector of function values at these points by $\bar{\mathbf{f}} := (f(\bar{\mathbf{x}}^1, \bar{\mathbf{p}}), \dots, f(\bar{\mathbf{x}}^N, \bar{\mathbf{p}}))^T \in \mathbb{R}^N$, and the vector of variables by $\mathbf{x} = (x_1, \dots, x_M)^T$. The RBF interpolation $S_\alpha : \mathbb{R}^M \mapsto \mathbb{R}$ of $f(\cdot, \bar{\mathbf{p}})$ is then defined (Wendland 2005, Section 6.3) as

$$S_\alpha(\mathbf{x}) := \sum_{n=1}^N \alpha_n \phi(\|\mathbf{x} - \bar{\mathbf{x}}^n\|) + \alpha_{N+1} + \sum_{m=1}^M \alpha_{N+1+m} x_m, \quad (2)$$

where ϕ denotes a given RBF, e.g., a linear, multi-quadratic, or cubic function, and α_n , $n = 1, \dots, N + 1 + M$, are the interpolation coefficients. The interpolation problem is that to find a vector $\alpha \in \mathbb{R}^{N+1+M}$ such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \alpha = \begin{pmatrix} \bar{\mathbf{f}} \\ \mathbf{0} \end{pmatrix}, \quad (3)$$

where $A_{ij} := \phi(\|\bar{\mathbf{x}}^i - \bar{\mathbf{x}}^j\|)$, $i, j = 1, \dots, N$, and $P_i := (1, (\bar{\mathbf{x}}^i)^T)$, $i = 1, \dots, N$.

The assignment (2) defines the unique RBF interpolation S_α of the unknown function $f(\cdot, \bar{\mathbf{p}})$ at the sample points $\bar{\mathbf{x}}^n$, $n = 1, \dots, N$, where the vector α is uniquely determined by

the system (3) of linear equations (Wendland 2005, Section 6.3). The RBF interpolation often yields good global representations of computationally expensive simulation-based functions, as judged by the errors estimated in Buhmann (2003, Chapter 5), and is therefore frequently used within RSMs.

3 Solving many instances of a simulation-based optimization problem with a partitioned decision space

We assume that the objective function f of (1) over the box defined by the constraints on the variables \mathbf{x} and the parameters \mathbf{p} is influenced significantly more by the variables than by the parameters, over the respective boxes. This assumption is characterized by the relation between the approximation errors, when f is approximated by 1st order Taylor expansions with respect to \mathbf{x} and \mathbf{p} , respectively, according to Assumption 1.²

Assumption 1 For the partition (\mathbf{x}, \mathbf{p}) , define

$$T_{\mathbf{x}} := \sup_{\mathbf{x}, \bar{\mathbf{x}}, \mathbf{p}} \left\{ \frac{|f(\mathbf{x}, \mathbf{p}) - f(\bar{\mathbf{x}}, \mathbf{p}) - D_{\mathbf{x}}f(\bar{\mathbf{x}}, \mathbf{p})^T(\mathbf{x} - \bar{\mathbf{x}})|}{\|\mathbf{x} - \bar{\mathbf{x}}\|} \mid \mathbf{x}, \bar{\mathbf{x}} \in [\mathbf{l}_{\mathbf{x}}, \mathbf{u}_{\mathbf{x}}], \mathbf{p} \in [\mathbf{l}_{\mathbf{p}}, \mathbf{u}_{\mathbf{p}}] \right\}$$

and

$$T_{\mathbf{p}} := \sup_{\mathbf{x}, \mathbf{p}, \bar{\mathbf{p}}} \left\{ \frac{|f(\mathbf{x}, \mathbf{p}) - f(\mathbf{x}, \bar{\mathbf{p}}) - D_{\mathbf{p}}f(\mathbf{x}, \bar{\mathbf{p}})^T(\mathbf{p} - \bar{\mathbf{p}})|}{\|\mathbf{p} - \bar{\mathbf{p}}\|} \mid \mathbf{x} \in [\mathbf{l}_{\mathbf{x}}, \mathbf{u}_{\mathbf{x}}], \mathbf{p}, \bar{\mathbf{p}} \in [\mathbf{l}_{\mathbf{p}}, \mathbf{u}_{\mathbf{p}}] \right\}.$$

Then, the relation $T_{\mathbf{x}} \gg T_{\mathbf{p}}$ is assumed to hold.

To identify the variables to be treated as parameters (i.e., the second argument, denoted \mathbf{p}) one can perform a simple parameter study or a sensitivity analysis; alternatively the parameters may be identified from literature on existing models of the specific computationally expensive function studied.

Each instance of the problem (1) is determined through a selection of values for the parameter vector \mathbf{p} . The optimization problem (1) will be solved to near-optimality for Q selected parameter values, denoted $\bar{\mathbf{p}}^q$, $q = 1, \dots, Q$. The other parameter settings, which are not known in advance and for which the problem (1) is to be (approximately) solved in a computationally efficient way, are denoted $\bar{\mathbf{p}}^r$, $r = Q + 1, \dots, Q + R$ (the value of R may be unknown). It is desirable that the degree of nonlinearity of f with respect to \mathbf{p} is low since it can be shown that if f is linear in \mathbf{p} , then its RBF interpolation S_{α} will be linear in \mathbf{p} as well. When f is almost linear with respect to \mathbf{p} the precision of the surrogate models for the parameter settings $\bar{\mathbf{p}}^r$, $r = Q + 1, \dots, Q + R$, will be as high as the precision of the surrogate models for $\bar{\mathbf{p}}^q$, $q = 1, \dots, Q$.

Section 3.1 introduces two standard solution methods for solving the set of optimization problems (1). These methods will be used as a benchmark for the proposed methodology described in Sect. 3.2.

3.1 Standard solution methods

Algorithm 2 is a standard solution method for the set of optimization problems (1) corresponding to the parameter settings $\bar{\mathbf{p}}^1, \dots, \bar{\mathbf{p}}^{Q+R}$. It applies an RSM based on an RBF interpolation

² In Assumption 1, the entities $D_{\mathbf{x}}f$ and $D_{\mathbf{p}}f$ denote numerical partial derivatives of f wrt. \mathbf{x} and \mathbf{p} respectively, computed, e.g., by forward or central differences (Allen 2008, Ch. 8)

(see, e.g., Jakobsson et al. 2010) of the function f that is valid for all $\mathbf{p} \in [\mathbf{l}_p, \mathbf{u}_p]$ and will be used as a benchmark for the proposed methodology.

Algorithm 2 An optimization algorithm based on response surfaces for a set of simulation-based optimization problem instances

- 0: Create a set of NQ sample points $(\bar{\mathbf{x}}^n, \bar{\mathbf{p}}^n)$, such that $(\bar{\mathbf{x}}^n, \bar{\mathbf{p}}^n) \in [\mathbf{l}_x, \mathbf{u}_x] \times [\mathbf{l}_p, \mathbf{u}_p]$, $n = 1, \dots, NQ$, and evaluate (simulate) f on this set.
 1: Construct a surrogate model S_α of f using the points evaluated, according to (2) and (3) with \mathbf{x} replaced by (\mathbf{x}, \mathbf{p}) .
 2: For $q = 1, \dots, Q + R$, find

$$\mathbf{x}_{\text{opt}}^q \in \underset{\mathbf{x} \in [\mathbf{l}_x, \mathbf{u}_x]}{\operatorname{argmin}} S_\alpha(\mathbf{x}, \bar{\mathbf{p}}^q). \quad (4)$$

The sample points created in Step 0 of Algorithm 2 are randomly generated in order to be uniformly distributed over the feasible set. We prioritize global search (exploration) when selecting the sample points in order to obtain a good global representation of the objective function, but any sophisticated rule for selecting sample points can be employed; see, e.g., Gutmann (2001, Algorithm 3). The total number of evaluations of f is upper bounded by the value NQ . For each instance $q \in \{1, \dots, Q + R\}$ of (1), the desired optimal solution $\mathbf{x}_{\text{opt}}^q$ is obtained by applying Algorithm 2.

Another natural approach is to first choose the values of $\bar{\mathbf{p}}^1, \dots, \bar{\mathbf{p}}^{Q+R}$, and then—for each parameters setting $\bar{\mathbf{p}}^q$ —call a standard algorithm for solving simulation-based optimization problems returning the optimal values of $\mathbf{x}_{\text{opt}}^q$, $q = 1, \dots, Q + R$. Such an approach allows the use of more sophisticated simulation-based optimization methods for both continuous and integer variables \mathbf{x} ; see Jones et al. (1998), Björkman and Holmström (2000) and Hong et al. (2015). We will use the easy to use software application for simulation-based optimization, NOMAD (Abramson et al. 2017), as the standard algorithm to find the optimal values of $\mathbf{x}_{\text{opt}}^q$. NOMAD is based on the Mesh Adaptive Direct Search (MADS) algorithm (Audet and Dennis 2006). A potential drawback of this approach is that only $NQ/(Q + R)$ sample points can be used to find each $\mathbf{x}_{\text{opt}}^q$, $q = 1, \dots, Q + R$, and that the value of $Q + R$ has to be known in advance. This approach also cannot be used to solve the tyre design problem, in which all the simulations of the objective function are required to be done in a preprocessing phase.

3.2 The proposed methodology

We propose Algorithm 3 below to solve a set of optimization problems (1). It includes a preprocessing phase, in which, for $\bar{\mathbf{p}}^q$, $q = 1, \dots, Q$, selected settings of the parameter vector, a nearly-optimal solution $\mathbf{x}_{\text{opt}}^q$ to (1) is computed provided that enough simulations of f can be performed. Then, for any other values of $\mathbf{p} \in [\mathbf{l}_p, \mathbf{u}_p]$, an approximately optimal solution $\mathbf{x}_{\text{approx}}^r$ is efficiently computed without any additional simulations of the expensive function f ; see Fig. 2. The desired solutions to the optimization problem (1) for all parameter settings are computed using the response surface method (Algorithm 1), i.e., a surrogate function is minimized instead of the true objective function f .

The sample points in Step 1a of Algorithm 3 are randomly generated to be uniformly distributed over the feasible set. We prioritize global search (exploration) when selecting the sample points to obtain a good global representation of the objective function, but any sophisticated rule for selecting sample points can be employed; see, e.g., Gutmann (2001,

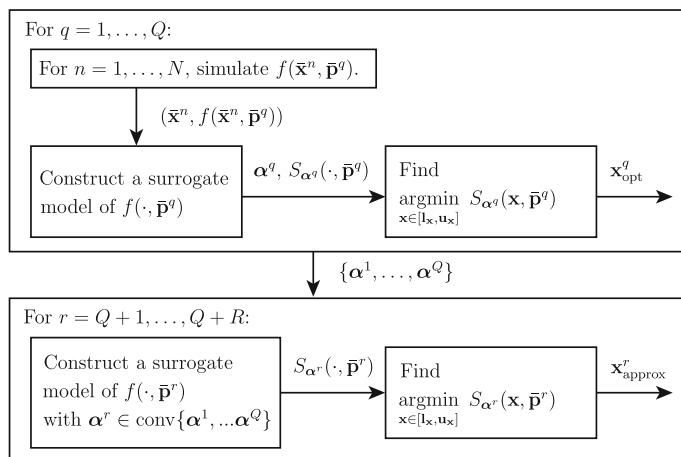


Fig. 2 Flowchart illustrating Algorithm 3 for the optimization of a set of simulation-based optimization problem instances

Algorithm 3 An efficient optimization algorithm based on weighted response surfaces for a set of simulation-based optimization problem instances

1: For each $q = 1, \dots, Q$:

- a: Select a distinct parameter setting $\bar{\mathbf{p}}^q \in [\mathbf{l}_{\mathbf{p}}, \mathbf{u}_{\mathbf{p}}]$. Create a set of N sample points $\bar{\mathbf{x}}^n$, such that $\bar{\mathbf{x}}^n \in [\mathbf{l}_{\mathbf{x}}, \mathbf{u}_{\mathbf{x}}]$, $n = 1, \dots, N$, and simulate $f(\bar{\mathbf{x}}^n, \bar{\mathbf{p}}^q)$, $n = 1, \dots, N$.
- b: Construct a surrogate model $S_{\alpha^q}(\cdot, \bar{\mathbf{p}}^q)$ according to (2) and (3) and find

$$\mathbf{x}_{\text{opt}}^q \in \underset{\mathbf{x} \in [\mathbf{l}_{\mathbf{x}}, \mathbf{u}_{\mathbf{x}}]}{\operatorname{argmin}} S_{\alpha^q}(\mathbf{x}, \bar{\mathbf{p}}^q). \quad (5)$$

2: For each $r = Q + 1, \dots, Q + R$:

- a: Construct a surrogate model $S_{\alpha^r}(\cdot, \bar{\mathbf{p}}^r)$ according to (2), letting the vector $\alpha^r \in \mathbb{R}^{N+1+D}$ of interpolation coefficients be a convex combination (e.g., assigned as in (8), below) of the initial vectors of coefficients, α^q , according to

$$\alpha^r := \sum_{q=1}^Q w_{rq} \alpha^q, \quad \sum_{q=1}^Q w_{rq} = 1, \quad w_{rq} \geq 0, \quad q = 1, \dots, Q. \quad (6)$$

- b: Compute an approximate optimal solution to (1), for $\mathbf{p} = \bar{\mathbf{p}}^r$:

$$\mathbf{x}_{\text{approx}}^r \in \underset{\mathbf{x} \in [\mathbf{l}_{\mathbf{x}}, \mathbf{u}_{\mathbf{x}}]}{\operatorname{argmin}} S_{\alpha^r}(\mathbf{x}, \bar{\mathbf{p}}^r). \quad (7)$$

Algorithm 3). The same set of sample points $\bar{\mathbf{x}}^n$, $n = 1, \dots, N$, is used for all surrogate models in Step 1a, since then the surrogate model $S_{\alpha^r}(\cdot, \bar{\mathbf{p}}^r)$ can be found directly by weighting the interpolation coefficients.³ The number of evaluations of the function f is limited to NQ , i.e., the same number as for Algorithm 2. The optimization problems (4) and (5) are solved

³ When using a different set of sample points for each parameter setting, the function values must instead be weighted and the interpolation coefficients have to be found by solving the system (3). The approach using different sets of sample points for different parameter settings results in the sample points being more spread over the search space. Experiments indicate that more accurate approximations $\mathbf{x}_{\text{approx}}^r$, $r = Q + 1, \dots, Q + R$, of the optimal solutions are obtained when using the latter approach.

because we are interested in near-optimal solutions to (1) for the selected parameter settings (the optimal tyre designs for the SVSs can then be used to point out the tyre designs to be improved with respect to the RRC, e.g., by adjusting their groove patterns).

The surrogate models $S_{\alpha^q}(\cdot, \bar{\mathbf{p}}^q) : \mathbb{R}^M \mapsto \mathbb{R}$ constructed in Step 1b of Algorithm 3 are defined on \mathbb{R}^M , since the second argument $\bar{\mathbf{p}}^q$ of the function f is fixed over the N sample points. The optimal solutions $\mathbf{x}_{\text{opt}}^q$ are not used further in the algorithm, but given to the practitioner. For the tyre design problem the optimal solutions $\mathbf{x}_{\text{opt}}^q$ are used to identify the strategic tyre designs to be further improved wrt. fuel efficiency.

Each of the surrogate models $S_{\alpha^r}(\cdot, \bar{\mathbf{p}}^r) : \mathbb{R}^M \mapsto \mathbb{R}$, $r = Q + 1, \dots, Q + R$, is constructed (in Step 2a of Algorithm 3) as a convex combination⁴ of the surrogate models $S_{\alpha^q}(\cdot, \bar{\mathbf{p}}^q)$, $q = 1, \dots, Q$. For each $r \in \{Q + 1, \dots, Q + R\}$, the values of the convex combination coefficients—here also referred to as *weights*— w_{rq} of α^q , in the surrogate model $S_{\alpha^r}(\cdot, \bar{\mathbf{p}}^r)$, are defined according to (6), and as a function of the values of $\bar{\mathbf{p}}^r$ and $\bar{\mathbf{p}}^q \in [\mathbf{l}_p, \mathbf{u}_p]$, $q = 1, \dots, Q$. Provided that $\bar{\mathbf{p}}^r \neq \bar{\mathbf{p}}^q$, $q = 1, \dots, Q$, the weights may, e.g., be defined as inversely proportional to the Euclidean distance between $\bar{\mathbf{p}}^r$ and $\bar{\mathbf{p}}^q$:

$$w_{rq} := \left(\|\bar{\mathbf{p}}^q - \bar{\mathbf{p}}^r\| \sum_{t=1}^Q \left(\frac{1}{\|\bar{\mathbf{p}}^t - \bar{\mathbf{p}}^r\|} \right) \right)^{-1}, \quad q = 1, \dots, Q. \quad (8)$$

In our test cases—for implementation simplicity—only the two parameter vectors $\bar{\mathbf{p}}^{q_1}$ and $\bar{\mathbf{p}}^{q_2}$ being closest⁵ to the parameter vector $\bar{\mathbf{p}}^r$, $r = Q + 1, \dots, Q + R$, are given nonzero weights, w_{rq_1} and w_{rq_2} , respectively. Several alternative principles for defining the weights w_{rq} may be employed. For example, they can be defined by more than two parameter vectors, higher weights can be assigned to surrogate models for values of \mathbf{p} for which the surrogates are expected to be more accurate, or a priori known properties of the function f (e.g., f being quadratic with respect to \mathbf{p}) can be utilized.

No additional simulations of the objective function f are needed to compute the approximately optimal solution $\mathbf{x}_{\text{approx}}^r$ of (7) in Step 2b of Algorithm 3. This constitutes a significant computational saving [as compared to solving (7) for each $\bar{\mathbf{p}}^r$ individually, which requires a new set of sample points to construct the surrogate model $S_{\alpha^r}(\cdot, \bar{\mathbf{p}}^r)$] and allows for the computation of approximately optimal solutions in real time.

The main advantage of Algorithm 3 as compared to Algorithm 2 is that the lower-dimensional box $[\mathbf{l}_x, \mathbf{u}_x] \subset \mathbb{R}^M$ is sampled for each of the selected settings of $\mathbf{p} \in [\mathbf{l}_p, \mathbf{u}_p] \subset \mathbb{R}^{D-M}$, instead of sampling the whole set $[\mathbf{l}_x, \mathbf{u}_x] \times [\mathbf{l}_p, \mathbf{u}_p] \subset \mathbb{R}^D$.

General RBF interpolating methods yield more and more accurate interpolations as new sample points are added. According to Buhmann (2003, Theorem 5.5), for locally Lipschitz continuous functions f , as the set of sample points grows dense in the domain, the surrogate function S_{α} converges to the true function f on this domain, which implies that the error estimate of the interpolation tends to zero. When considering a simulation-based function f we cannot presume that it is locally Lipschitz continuous and the performance of Algorithm 3 has to be assessed through computational experiments.

⁴ The use of a convex combination is motivated by (i) the assumption that the objective function f is influenced significantly more by the variables than by the parameters, and (ii) the assumption that the parameter values $\bar{\mathbf{p}}^q$, $q = 1, \dots, Q$, are spread over the box $[\mathbf{l}_p, \mathbf{u}_p]$. If f is linear in \mathbf{p} , then the RBF interpolation S_{α^r} is equal to the convex combination of S_{α^q} , $q = 1, \dots, Q$.

⁵ $q_1 \in \operatorname{argmin}_{q \in \{1, \dots, Q\}} \{\|\bar{\mathbf{p}}^q - \bar{\mathbf{p}}^r\|\}$ and $q_2 \in \operatorname{argmin}_{q \in \{1, \dots, Q\} \setminus \{q_1\}} \{\|\bar{\mathbf{p}}^q - \bar{\mathbf{p}}^r\|\}$.

4 Computational experiments

The purpose of the computational experiments is to compare the performance of Algorithms 2 and 3 and NOMAD applied to a set of simulation-based optimization problems. The test problems, whose properties resemble those of the tyre design problem that we wish to solve, were collected from Gould et al. (2003) and Montaz Ali et al. (2005) and are often used for testing optimization algorithms aiming at convergence to a global optimum.

The methodology used for assessing the performance of the algorithms tested, as well as the performance measures used, are introduced in Sect. 4.1. The test problems are described in Sect. 4.2. Section 4.3 discusses the implementation details, while Sect. 4.4 summarizes the test results. Section 4.5 describes how the tyre design problem can be solved using the algorithm developed.

4.1 Performance measures and methodology

This section presents the *success ratio*, designated to compare the effectiveness and the trustworthiness of algorithms, as well as the *performance profile* and *data profile*, designated for more complex comparisons of algorithms.

Success ratio

Letting \mathcal{A} be the set of tested algorithms and \mathcal{P} be the set of test problems, the success ratio $s_{ap} \in [0, 1]$ equals the number of times that algorithm $a \in \mathcal{A}$ successfully approximates an optimum to the problem $p \in \mathcal{P}$ divided by the number of times that the algorithm is applied to solve the problem p (Törn and Žilinskas 1989, Ch. 1).

Consider the instance r of problem p of the form (1) defined by $\mathbf{p} = \bar{\mathbf{p}}^r$, and let $X_{\text{opt}}^r \subset [\mathbf{l}_x, \mathbf{u}_x] \subset \mathbb{R}^M$ denote its optimal set. An algorithm $a \in \mathcal{A}$ is said to successfully approximate its optimal solution if the resulting point is feasible and lies in a sufficiently small neighbourhood of the set of optimal solutions, i.e., if $\mathbf{x}_{\text{approx}}^r$ belongs to the set

$$N_{\varepsilon_p}(X_{\text{opt}}^r) := \left\{ \mathbf{x} \in [\mathbf{l}_x, \mathbf{u}_x] \mid \min_{\mathbf{y} \in X_{\text{opt}}^r} \|\mathbf{x} - \mathbf{y}\| \leq \varepsilon_p \right\}, \quad (9)$$

where $\varepsilon_p > 0$ is a tolerance parameter. For each problem p the value of ε_p is determined by the size and dimension of the box $[\mathbf{l}_x, \mathbf{u}_x]$ and is reported in “Appendix”.

For all the algorithms tested, the same number of points are sampled. In Algorithm 2, all the sample points are used to map a subspace of \mathbb{R}^D , while in Algorithm 3 the number of sample points is divided, so that half of them are used to sample a subspace of \mathbb{R}^M (where $M < D$) for each of the two selected parameter vectors, i.e., $Q = 2$. When NOMAD is used the number of sample points is divided as well: two thirds of them are used to sample the subspaces \mathbb{R}^M for each of the two selected parameter vectors, and one third is used to solve the problem for $Q + 1$, i.e., to find $\mathbf{x}_{\text{approx}}^{Q+1}$.

Performance and data profile

The *performance profile* introduced in Dolan and Moré (2002) has been utilized for comparing the performance of Algorithms 2 and 3 on the whole set of test problems considered simultaneously. The performance profile for an algorithm is defined as the (cumulative) distribution function for a performance metric, here represented by the number of sample points. Assuming a set \mathcal{A} of algorithms applied to a set \mathcal{P} of problems, for each $p \in \mathcal{P}$ and $a \in \mathcal{A}$,

t_{pa} is defined as the number of sample points used to successfully approximate (in the variable space) an optimum of the problem p by algorithm a . The *performance ratio*, defined as $t_{pa}(\min_{b \in \mathcal{A}}\{t_{pb}\})^{-1}$, $p \in \mathcal{P}$, $a \in \mathcal{A}$, relates the performance of algorithm a as applied to the problem p to the best performance of any of the algorithms in the set \mathcal{A} . If algorithm a is not able to successfully approximate an optimum to the problem p (i.e., if $t_{pa} > T$ for some large value $T > 0$), then the performance ratio is set to a constant $c_{ap} > \max_{b \in \mathcal{A} \setminus \{a\}}\{t_{pb}\}$. An overall assessment of the performance of algorithm a is obtained by defining the probability that the performance ratio is within a factor $\tau \geq 1$ of the best attained ratio for any of the algorithms, i.e.,

$$\rho_a(\tau) := \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{t_{pa}}{\min_{b \in \mathcal{A}}\{t_{pb}\}} \leq \tau \right\} \right|,$$

where $|\cdot|$ denotes cardinality. The function $\rho_a : [1, \infty) \mapsto [0, 1]$ is the cumulative distribution function of the performance ratio.

A plot of the performance profile for the set \mathcal{A} of tested algorithms reveals the major performance characteristics of the algorithm but does not provide sufficient information to a user with a computationally expensive optimization problem, who is often more interested in the performance of solvers as functions of the number of function evaluations. We thus use the *data profile* of an algorithm $a \in \mathcal{A}$ introduced in Moré and Wild (2009), defined as the percentage of the problems that are solved at the cost of $\tau > 0$ with n_p being the number of variables of problem $p \in \mathcal{P}$, i.e.,

$$d_a(\tau) := \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{t_{pa}}{n_p + 1} \leq \tau \right\} \right|.$$

The unit of cost is $n_p + 1$ function evaluations which can be easily translated into function evaluations. This unit can be interpreted as the percentage of problems that can be solved with the equivalent of simplex gradient estimates, $n_p + 1$ referring to the number of evaluations needed to compute a one-sided finite-difference estimate of the gradient.

Performance profiles compare the different algorithms, while data profiles provide, for each given algorithm (independent of the other algorithms), the proportion of the problems that are solved within a certain number of function simulations normalized by the corresponding number of variables by a given algorithm. We will present both performance and data profiles for the test problems considered.

Assessment methodology

The overall methodology used to assess and compare the performance of Algorithms 2 and 3 is described next. The sample points $\bar{\mathbf{x}}^n$, $n = 1, \dots, NQ$, were randomly generated to be uniformly distributed over the feasible sets. Also the latin hypercube design of experiments (Sóbestor et al. 2014) was tested to generate the sample points, but since the resulting improvements of the success of approximation of an optimal solution—as compared with randomly generated sample points—was negligible we chose to use sample points with randomly generated coordinates over their respective feasible intervals (using Matlab's pseudo-random number generator). Considering the various measures for comparing algorithms discussed in this section, the following experimental procedure, inspired by Montaz Ali et al. (2005), was established.

1. Assemble a set \mathcal{P} of test problems from the literature. For each of the problems chosen, record the variables and objective values at an optimal solution (or at the best known solution).
2. Choose the number NQ of sample points (i.e., $\bar{\mathbf{x}}^n$) to create. For each problem: perform 30 runs each of Algorithms 2 and 3.
3. For each problem p and each algorithm a , record the success ratio s_{ap} .
4. Increase the number NQ of sample points. Repeat from step 2 until either Algorithm 2 or Algorithm 3 successfully approximates an optimum for each problem p .⁶
5. Generate the performance and data profiles.

The success ratio is used when repeated runs of an optimization algorithm may give different results, which is the case for Algorithms 2 and 3, which use randomly generated sample points. When using NOMAD the sample points are deterministically generated, creating a mesh. Hence, it is enough to apply NOMAD once per problem and therefore its success ratio will not be reported.

4.2 Test problems

The experimental procedure described in Sect. 4.1 has been applied to 22 selected box-constrained optimization problems with continuous variables; these are denoted $p \in \{1, \dots, 22\}$ below. These problems (except for $p \in \{6, 7, 12, 16, 20, 21\}$) form a subset of a test-bed for global optimization solvers originally proposed in Montaz Ali et al. (2005), and include both artificial and real problems; see Table 1 and “Appendix”. To make the tests demonstrative, problems differing in dimension, computational difficulty, and number of known local optima are selected. Most of them are considered as fairly easy in terms of global optimization; when viewed as simulation-based optimization problems—in which normally no analytical information can be exploited—they are, however, challenging.

The proposed algorithm is developed for optimization problems in which the influence of some of the variables on the value of the objective function is less significant than that of the rest of the variables and one wants to solve many instances of the problem. The less influencing variables are then treated as parameters, denoted by the vector \mathbf{p} . From the 50 test problems proposed in Montaz Ali et al. (2005) in fifteen problems (1, 3–5, 8–11, 13–15, 17–19, and 22) the less influencing variables are identified based on the definition of significant influence in Assumption 1 and will be used to assess the performance of Algorithm 3. Problem 2 is included in order to demonstrate the performance of the algorithm when applied to a problem for which the objective function is influenced equally by all its variables. The Problems 6, 7, 12, 16, 20, and 21 (taken from Schwefel 1981; Bersini et al. 1996; Dixon and Price 1999; Gould et al. 2003; Rahnamyan et al. 2007, and gathered in Jamil and Yang 2013) are considered in order to test the algorithms for a larger number of variables, for which Algorithm 3 is intended.

The tyre design problem in the current implementation contains four variables and 55 parameters; hence, its dimensions D and M are similar to Problem 7. The results from Algorithm 3 applied to this problem will be commented on below.

⁶ See the definition of $N_{\varepsilon_p}(X_{\text{opt}}^r)$ in (9), where the values of the tolerance parameter $\varepsilon_p > 0$ differs between the test problems but not between the algorithms.

Table 1 Data for the test problems: problem number (p), problem name, problem dimension ($\mathbf{x} \in \mathbb{R}^M$, $\mathbf{p} \in \mathbb{R}^{D-M}$), and partition into variables \mathbf{x} and parameters \mathbf{p}

p	Name	D	M	\mathbf{x}	\mathbf{p}
1	Aluffi-Pentini	2	1	x_1	x_2
2	Becker and Lago	2	1	x_1	x_2
3	Branin	2	1	x_1	x_2
4	Camel back—three hump	2	1	x_1	x_2
5	Camel back—six hump	2	1	x_1	x_2
6	Dixon and Price	20	19	x_1, \dots, x_{19}	x_{20}
7	Explin	120	11	x_1, \dots, x_{11}	x_{12}, \dots, x_{120}
8	Gulf research	3	2	x_2, x_3	x_1
9	Helical valley	3	2	x_1, x_2	x_3
10	Hosaki	2	1	x_2	x_1
11	Kowalik	4	3	x_2, x_3, x_4	x_1
12	Langerman	10	5	x_1, \dots, x_5	x_6, \dots, x_{10}
13	Meyer and Roth	3	2	x_1, x_2	x_3
14	Miele and Cantrell	4	3	x_1, x_3, x_4	x_2
15	Modified Rosenbrock	2	1	x_1	x_2
16	Powell	20	10	x_{11}, \dots, x_{20}	x_1, \dots, x_{10}
17	Prices's transistor	9	7	$x_1, x_2, x_5, \dots, x_9$	x_3, x_4
18	Rosenbrock—2-dim.	2	1	x_1	x_2
19	Rosenbrock— n -dim.	10	9	x_1, \dots, x_9	x_{10}
20	Schwefel	20	10	x_1, \dots, x_{10}	x_{11}, \dots, x_{20}
21	Tointqor	50	46	x_1, \dots, x_{46}	x_{47}, \dots, x_{50}
22	Wood	4	2	x_1, x_3	x_2, x_4

See “Appendix” for detailed descriptions of the problems and the corresponding references

4.3 Implementation

We implemented the efficient optimization algorithm for a set of simulation-based problem instances (Algorithm 3) and the standard optimization algorithm based on an RBF interpolation (Algorithm 2) in MATLAB R2010b (The MathWorks, Inc. 2012). All experiments were carried out on a desktop computer equipped with Intel Pentium Dual-Core 2.80 GHz and 4 GB RAM, running Red Hat Enterprise Linux 5.6. To optimize the nonlinear interpolation functions S_α generated during the various steps of the algorithms, subject to box constraints, the external solver glbFast from the TOMLAB/CGO v8.0 toolbox for global optimization (Holmström and Göran 2002) was used; it solves box-constrained global optimization problems using only function values, and is based on the DIRECT algorithm from Jones et al. (1993). In our implementation of Algorithms 2 and 3 the simple linear RBF $g : \mathbb{R}^D \mapsto \mathbb{R}_-$, defined by $g(\mathbf{x}) = \phi(\|\mathbf{x}\|) := -\|\mathbf{x}\|$, was employed in order to construct the RBF interpolations (2). We have allowed NOMAD to use Latin-Hypercube Search and Variable Neighbourhood Search when it got stuck in a local optimum, in order to improve its performance and use the same number of sample points as the other algorithms tested.

Table 2 Success ratios for Algorithm 2 (denoted alg2) and Algorithm 3 (denoted alg3)

p	$s_{\text{alg2},p}$	$s_{\text{alg3},p}$
1	0.20	1.00
2	0.27	0.37
3	0.07	0.10
4	0.03	0.80
5	0.07	0.97
6	0.50	0.70
7	0.43	0.67
8	0.10	0.20
9	0.37	0.40
10	0.47	0.77
11	0.20	0.27
12	0.23	0.34
13	0.13	0.03
14	0.07	0.07
15	0.17	1.00
16	0.40	0.47
17	0.30	0.33
18	0.37	1.00
19	0.07	0.20
20	0.10	0.13
21	0.00	0.30
22	0.17	0.13

4.4 Results

A general optimization algorithm converges to the global optimum for every continuous objective function f if and only if the set of sample points grows dense over the feasible domain; see Gutmann (2001, Theorem 4). Nevertheless, the determination of an acceptable result after a reasonably small number of function evaluations is often its most desirable feature. Therefore, the quality of our optimization algorithm for simulation-based functions is demonstrated through its practical use for solving representative test problems utilizing relatively few function evaluations.

The success ratios for Algorithms 2 and 3—possessing values in the interval $[0, 1]$ —are listed in Table 2. Each bold entry indicates the algorithm that performs the best when applied to the respective test problem.

In Figs. 3 and 4, the performance profiles (Dolan and Moré 2002) and data profiles (Moré and Wild 2009) of Algorithms 2 and 3 and NOMAD over the 22 test problems considered are illustrated. Algorithm 3, which is proposed in this paper, clearly outperforms both NOMAD and Algorithm 2 (being a version of Algorithm 1, adjusted in order to suit the specific problem considered, but which does not utilize the partition (\mathbf{x}, \mathbf{p})) for the problems tested. In the worst case Algorithm 3 needed approximately twice as many sample points as the other algorithms. Figures 5 and 6, respectively, illustrate the performance and data profiles of Algorithms 2 and 3 and NOMAD over the larger test problems considered, i.e., (6, 7, 12, 16, 17, and 19–21). Algorithm 3, developed for solving a set of simulation-based optimization problem

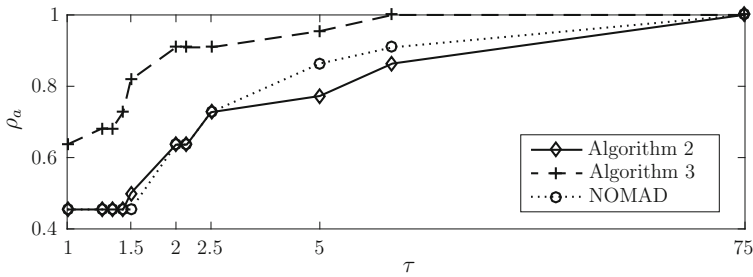


Fig. 3 Performance profiles of Algorithms 2 and 3 and NOMAD for the 22 problems tested. The graphs represent the portion ρ_a of the problems that are successfully approximated (see Sect. 4.1; the value of the tolerance parameter ε_p varies among the test problems) within a factor $\tau \in \{1, 1.25, 1.33, 1.43, 1.5, 2, 2.15, 2.5, 5, 8, 75\}$ times the number of sample points used by the algorithm that successfully approximates an optimal solution using the smallest number of sample points. The scale used for the horizontal axis is logarithmic

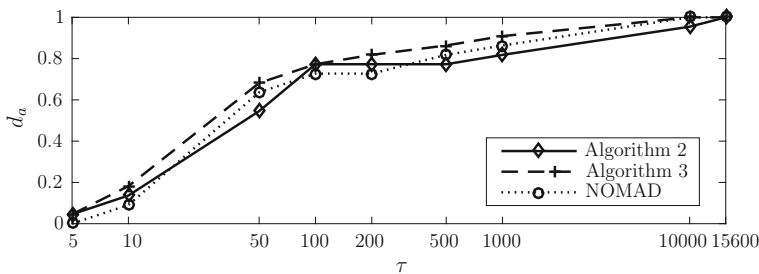


Fig. 4 Data profiles of Algorithms 2 and 3 and NOMAD for the 22 problems tested. The graphs represent the portion d_a of the problems that are successfully approximated (see Sect. 4.1; the value of the tolerance parameter ε_p varies among the test problems) within $\tau(n_p + 1) \in \{5, 10, 50, 100, 200, 500, 1000, 10,000, 15,600\}$ sample points used. The scale used for the horizontal axis is logarithmic

instances in a computationally efficient way, turned out to be particularly suitable for the larger problems considered.

We next discuss the cases when the algorithm developed performed worse than the standard RSM and NOMAD:

- Both Algorithms 2 and 3 terminate at inaccurate approximations of the global optimum of Problem 13; this is probably because the objective function decreases steeply towards the global minimum in a small area, which is difficult to locate with only a small number of randomly generated sample points. Hence, another (more sophisticated) rule for selecting sample points should be employed. NOMAD outperforms both Algorithms 2 and 3 on Problem 13 because new sample points are generated based on the function values at the old sample points.
- Both Algorithms 2 and 3 find inaccurate estimations of the optimal objective value of Problem 17, because the objective function is oscillating around the global optimum. A more suitable surrogate model should be chosen for Problem 17 in order to obtain more accurate estimations of the optimal objective value. NOMAD is not based on any surrogate model so the estimation of the optimal objective value of Problem 17 is more accurate but corresponds to a less accurate approximation of the global optimum.

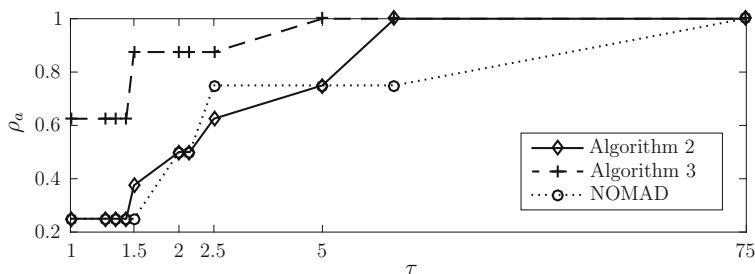


Fig. 5 Performance profiles of Algorithms 2 and 3 and NOMAD for the larger problems (6, 7, 12, 16, 17, and 19–21) tested. The graphs represent the portion ρ_a of the problems that are successfully approximated within a factor $\tau \in \{1, 1.25, 1.33, 1.43, 1.5, 2, 2.15, 2.5, 5, 8, 75\}$ times the number of sample points used by the algorithm that successfully approximates an optimal solution using the smallest number of sample points. The scale used for the horizontal axis is logarithmic

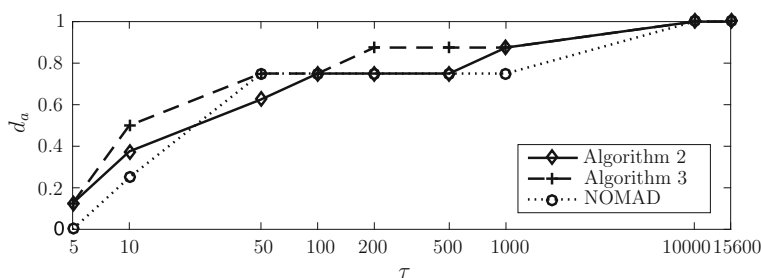


Fig. 6 Data profiles of Algorithms 2 and 3 and NOMAD for larger problems (6, 7, 12, 16, 17, and 19–21) tested. The graphs represent the portion d_a of the problems that are successfully approximated (see Sect. 4.1; the value of the tolerance parameter ε_p varies among the test problems) within $\tau(n_p + 1) \in \{5, 10, 50, 100, 200, 500, 1000, 10,000, 15,600\}$ sample points used. The scale used for the horizontal axis is logarithmic

- Even though it is not possible to decide which variable should be treated as a parameter in Problem 2 (the objective function is quadratically influence by each of the variables), Algorithm 3 yields more accurate results than Algorithm 2.

The performance of NOMAD increases with an increasing number of sample points used. When solving a large number of instances of (1) using only a limited number of sample points, NOMAD will lead to inaccurate approximations of the global optima; see Figs. 3 and 4.

We compared the approximate solutions to the test problems resulting from partitions of the decision variables (into the variables \mathbf{x} and the parameters \mathbf{p}), being either randomly selected or based on a priori information about the objective function. Algorithm 3 produces more accurate approximations of the optimal solutions for the partitions based on a priori information than for the randomly selected partitions. For an equal number of sample points, the approximate solutions computed by the computationally efficient Algorithm 3 are in most cases still more accurate than the ones resulting from the application of Algorithm 2.

Problem 7, for which it is obvious from the explicit expression of the objective function how to choose the variables $\mathbf{x} \in \mathbb{R}^M$ and parameters $\mathbf{p} \in \mathbb{R}^{D-M}$, was selected in order to investigate how the relative sizes of the sets of variables/parameters influence the performance of Algorithm 3. The accuracy of the approximately optimal solution obtained does not appear

to be influenced by the ratio $\frac{M}{D}$ for the same number of sample points, whereas the lowest value of the objective function is reduced when this ratio is decreased. From this we conclude that as many variables as possible not significantly influencing the objective function should be treated as parameters in Algorithm 3.

4.5 Application to the tyre design problem

Our aim is to apply Algorithm 3 to solve the tyre design problem in the combinatorial setting of vehicles and operating environments. The surrogate models of the objective function and the corresponding optimal tyre designs—for the strategic configurations of vehicles operating in the most common environments—are found in a pre-processing phase involving many computationally expensive simulations of the objective function (representing the energy losses caused by the tyres). An approximately optimal tyre design can then be found in a computationally efficient way (and which then can be implemented in the sales tool at Volvo) for any customer specifying the vehicle and its intended use, by using a weighted surrogate model of the objective function, as described in this article. The tyre design variables are so far considered to be continuous. We are currently developing the algorithm to incorporate also discrete requirements on the variable values. The algorithm developed can, however, be directly used for both continuous and discrete requirements on the parameters \mathbf{p} ; the individual vehicles in the tyre design problem then correspond to discrete values of the parameters.

A test instance of the tyre design problem is presented below. A Volvo rigid truck with four wheels with the two rear wheels driven is considered. The tyres are the same within each axle but can differ between the front and rear axles. Each tyre is determined by the values of three continuous variables (the tyre diameter, the tyre width, and the inflation pressure), resulting in a simulation-based optimization problem with the variable vector $\mathbf{x} = (x_1, \dots, x_6)^T \in \mathbb{R}^6$. Upper ($\mathbf{u}_x = (u_{x_1}, \dots, u_{x_6})^T$) and lower ($\mathbf{l}_x = (l_{x_1}, \dots, l_{x_6})^T$) bounds on the variables result in box constraints. Each evaluation of the objective function, which represents the energy losses caused by the tyres, requires a simulation of the vehicle over its operating environment; see Kolář (2015) for a description of the joint vehicle, tyres, and operating environment model used. The energy losses to be minimized are measured in MJ/km.

The parameter $\mathbf{p} \in \mathbb{R}$ which will be varied over the instances of the tyre design problem represents the topography of the operating environment. The other parameters, describing the truck, the operating environment, and the tyres, are kept fixed. The SVSs include the truck operating on a flat road and the same truck operating on a hilly road. Approximately optimal tyre designs (for both the front and rear axles) for the truck operating on a predominantly flat road are to be found by Algorithm 3 when N sample points are used for each of the SVSs. The approximately optimal tyre designs are compared with the optimal tyre designs found when the tyre design problem for the truck operating on a predominantly flat road is solved directly using $2N$ sample points; see Table 3.

The optimal solution to the tyre design problem for the truck operating on a predominantly flat road was found to be $(l_{x_1}, l_{x_2}, u_{x_3}, l_{x_4}, l_{x_5}, u_{x_6})$. Algorithm 3 successfully approximates the optimal designs while being computationally efficient when sufficiently many sample points can be used in the pre-processing phase for the strategic configurations operating in the most common environments, as can be seen in Table 3. This fact does not cause any difficulties with the implementation of Algorithm 3 into the sales tool where the computational efficiency of finding the approximately optimal tyre designs for any customer is important. The optimal solution to the tyre design problem for the truck operating on a predominantly flat road was found to be a combination of the lower and upper bounds on the variable values. Therefore, additional constraints, such as the handling quality, the ride comfort, and the

Table 3 The approximately optimal tyre designs $\mathbf{x}_{\text{approx}}^r$ for the truck operating on a predominantly flat road, and the corresponding energy losses $S_{\alpha^r}(\mathbf{x}_{\text{approx}}^r)$ (MJ/km), when N sample points are used

N	$\mathbf{x}_{\text{approx}}^r$	$S_{\alpha^r}(\mathbf{x}_{\text{approx}}^r)$	\mathbf{x}_{opt}	$S_{\alpha}(\mathbf{x}_{\text{opt}})$
50	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.88	$(l_{x1}, l_{x2}, l_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.84
100	$(l_{x1}, l_{x2}, l_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.47	$(l_{x1}, l_{x2}, l_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.86
150	$(l_{x1}, l_{x2}, l_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.49	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.89
200	$(l_{x1}, l_{x2}, l_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.48	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.90
250	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.50	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.91
300	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.51	$(l_{x1}, l_{x2}, u_{x3}, l_{x4}, l_{x5}, u_{x6})$	4.91

The optimal tyres designs found when the tyres design problem for the truck operating on a predominantly flat road is solved directly with $2N$ sample points and the corresponding energy losses $S_{\alpha}(\mathbf{x}_{\text{opt}})$ (MJ/km) are listed for comparison

startability of the truck should be added to the optimization model of the tyre design problem and the simulation model of the objective function should be improved.

5 Conclusions and future research

A computationally efficient optimization algorithm—based on a radial basis function interpolation adapted to a large set of simulation-based problem instances—has been developed, implemented, and tested. Our computational results demonstrate a performance of the algorithm which is superior to that of a standard response surface method (RSM) as well as NOMAD, when applied to a standard set of test problems. The algorithm is particularly suitable when the differentiation of parameters and variables can be distinctly resolved, and when aiming to solve efficiently the optimization problem at hand—at least approximately—for a large number of parameter settings. We consider problem settings possessing box constraints on the variables. The algorithm proposed can, however, be extended to solve optimization problems with more general (even simulation-based) constraints, then being relaxed and replaced by penalty terms in the objective function.

To compose surrogate models for any settings of the parameters, the algorithm utilizes aggregated surrogate functions of a set of selected parameter settings. Other models for the aggregation of the surrogate functions of the selected parameter settings should be investigated. The algorithm may also be generalized to be applicable in the context of utilizing other kinds of surrogate models (e.g., Kriging interpolation or polynomial regression).

The algorithm developed enables the computationally efficient solution of the truck tyre design problem when described in the combinatorial domain of possible vehicle configurations and operating environment specifications. Our future research includes incorporating discrete requirements on the variable values into the algorithm developed. This will enable the solution of a true tyres selection problem, in which the feasible tyres are taken from a tyre database, resulting in practically more useful solutions in place of the tyre designs resulting from the problem considered in this paper.

Acknowledgements The research leading to the results presented in this paper was financially supported by the Swedish Energy Agency (Project Number P34882-1), Chalmers University of Technology, and Volvo Group Trucks Technology. We acknowledge three anonymous referees for their valuable comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix: Global optimization test problems

We present the 22 box-constrained test problems selected among problems that are frequently used for testing global optimization algorithms. Most of these problems are presented in Montaz Ali et al. (2005); however, the problems 6, 7, 12, 16, 20, and 21 are taken from Schwefel (1981), Bersini et al. (1996), Dixon and Price (1999), Gould et al. (2003) and Rahnamyan et al. (2007), respectively. The problems selected are suitable for testing the algorithm presented because the significance of the influence on the function values differs between subsets of the variables. The only exception is Problem 2, which is selected in order to investigate whether the algorithm can be successful also when all of the variables influence the objective function to a similar extent. The tolerance parameter $\varepsilon_p > 0$ is stated for each problem p .

1. *Aluffi-Pentini problem* (Aluffi-Pentini et al. 1985):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := 0.25x_1^4 - 0.5x_1^2 + 0.1x_1 + 0.5x_2^2,$$

subject to $x_i \in [-10, 10]$, $i = 1, 2$. The problem has two local minima; one is global, located at $(-1.0465, 0)^T$, with the optimal objective value $f^* \approx -0.3523$. The tolerance parameter value is $\varepsilon_1 := 0.05$.

2. *Becker and Lago problem* (Price 1983):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := (\|x_1\| - 5)^2 + (\|x_2\| - 5)^2,$$

subject to $x_i \in [-10, 10]$, $i = 1, 2$. The function has four global minima, located at $(\pm 5, \pm 5)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_2 := 0.025$.

3. *Branin problem* (Dixon and Szegö 1978):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := a(x_2 - bx_1^2 + cx_1 - d)^2 + g(1 - h) \cos(x_1) + g,$$

where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $g = 10$, $h = 1/(8\pi)$, subject to $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$. The global minima are located at $(-\pi, 12.275)^T$, $(\pi, 12.275)^T$, and $(3\pi, 12.275)^T$, with $f^* = 5\pi/4$. The tolerance parameter value is $\varepsilon_3 := 0.025$.

4. *Camel back—three-hump problem* (Dixon and Szegö 1978):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 + x_1x_2 + x_2^2,$$

subject to $x_i \in [-5, 5]$, $i = 1, 2$. The problem has three local minima; one of them is global, located at $(0, 0)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_4 := 0.01$.

5. *Camel back—six hump problem* (Dixon and Szegö 1978):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

subject to $x_i \in [-5, 5]$, $i = 1, 2$. The objective function is symmetric around the origin and has three conjugate pairs of local minima, two of them are global, located at $\pm (0.089842, -0.712656)^T$, with $f^* \approx -1.0316$. The tolerance parameter value is $\varepsilon_5 := 0.01$.

6. *Dixon and Price problem* (Dixon and Price 1999):

$$\underset{\mathbf{x} \in \mathbb{R}^{20}}{\text{minimize}} \quad f(\mathbf{x}) := (x_1 - 1)^2 + \sum_{i=2}^{20} i(2x_i^2 - x_{i-1})^2,$$

subject to $x_i \in [0, 10]$, $i = 1, \dots, 20$. The global minimum is located at $x_i^* = 2^{-\frac{2^i-2}{2^i}}$, $i = 1, \dots, 20$, with $f^* = 0$. The tolerance parameter value is $\varepsilon_6 := 0.05$.

7. *Explin problem* (Gould et al. 2003):

$$\underset{\mathbf{x} \in \mathbb{R}^{120}}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^{10} \exp(0.1x_i x_{i+1}) + \sum_{i=1}^{120} (-10ix_i),$$

subject to $x_i \in [-10, 10]$, $i = 1, \dots, 120$. The best known solution, which can be found in Shcherbina et al. (2003), with $f(\mathbf{x}) = -723756.3$, was obtained by the solver LINGO8 (Ding et al. 2009). The tolerance parameter value is $\varepsilon_7 := 10$.

8. *Gulf research problem* (Moré et al. 1981):

$$\underset{\mathbf{x} \in \mathbb{R}^3}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^{99} \left[\exp\left(-\frac{(u_i - x_2)^{x_3}}{x_1}\right) - 0.01i \right]^2,$$

where $u_i = 25 + [-50 \ln(0.01i)]^{1/1.5}$, subject to $x_1 \in [0.1, 100]$, $x_2 \in [0, 25.6]$, and $x_3 \in [0, 5]$. The global minimum is located at $(50, 25, 1.5)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_8 := 0.25$.

9. *Helical valley problem* (Wolfe 1978):

$$\underset{\mathbf{x} \in \mathbb{R}^3}{\text{minimize}} \quad f(\mathbf{x}) := 100 \left[(x_2 - 10\theta(\mathbf{x}))^2 + (\sqrt{(x_1^2 + x_2^2)} - 1)^2 \right] + x_3^2,$$

where

$$\theta(\mathbf{x}) = \begin{cases} \frac{1}{2\pi} \tan^{-1} \frac{x_2}{x_1}, & \text{if } x_1 \geq 0, \\ \frac{1}{2\pi} \tan^{-1} \frac{x_2}{x_1} + \frac{1}{2}, & \text{if } x_1 < 0, \end{cases}$$

subject to $x_i \in [-10, 10]$, $i = 1, 2, 3$. The objective function is a steep-sided valley which follows a helical path. The global minimum is located at $(1, 0, 0)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_9 := 0.25$.

10. *Hosaki problem* (Benke and Skinner 1991):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := (1 - 8x_1 + 7x_1^2 - 7/3x_1^3 + 1/4x_1^4)x_2^2 \exp(-x_2),$$

subject to $x_1 \in [0, 5]$, $x_2 \in [0, 6]$. There are two minima, of which the global minimum is located at $(4, 2)^T$ with $f^* \approx -2.3458$. The tolerance parameter value is $\varepsilon_{10} := 0.01$.

11. *Kowalik problem* (Jansson and Knüppel 1995) (a least squares problem):

$$\underset{\mathbf{x} \in \mathbb{R}^4}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^{11} \left[a_i - \frac{x_1(1 + x_2b_i)}{(1 + x_3b_i + x_4b_i^2)} \right]^2,$$

subject to $x_i \in [0, 0.42]$, $i = 1, 2, 3, 4$. The values of a_i and b_i are given in Table 4. The global minimum is located at $(0.192, 0.190, 0.123, 0.135)^T$ with $f^* \approx 3.0748 \times 10^{-4}$. The tolerance parameter value is $\varepsilon_{11} := 0.05$.

Table 4 Data for the Kowalik problem

i	1	2	3	4	5	6	7	8	9	10	11
a_i	0.1957	0.1947	0.1735	0.16	0.0844	0.0627	0.0456	0.0342	0.0323	0.0235	0.0246
b_i	0.25	0.50	1.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0

Table 5 Data for Langerman problem

i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}	a_{i8}	a_{i9}	a_{i10}	c_i
1	9.681	0.667	4.783	9.095	3.517	9.325	6.544	0.211	5.122	2.020	0.806
2	9.400	2.041	3.788	7.931	2.882	2.672	3.568	1.284	7.033	7.374	0.517
3	8.025	9.152	5.114	7.621	4.564	4.711	2.996	6.126	0.734	4.982	1.500
4	2.196	0.415	5.649	6.979	9.510	9.166	6.304	6.054	9.377	1.426	0.908
5	8.074	8.777	3.467	1.863	6.708	6.349	4.534	0.276	7.633	1.567	0.965

Table 6 Data for the Meyer and Roth problem

i	t_i	v_i	y_i
1	1.0	1	0.126
2	2.0	1	0.219
3	1.0	2	0.076
4	2.0	2	0.126
5	0.1	0	0.186

12. *Langerman problem* (Bersini et al. 1996):

$$\underset{\mathbf{x} \in \mathbb{R}^{10}}{\text{minimize}} \quad f(\mathbf{x}) := - \sum_{i=1}^5 c_i \exp \left(- \frac{1}{\pi} \sum_{j=1}^{10} (x_j - a_{ij})^2 \right) \cos \left(\pi \sum_{j=1}^{10} (x_j - a_{ij})^2 \right),$$

subject to $x_j \in [0, 10]$, $j = 1, \dots, 10$. The global minimum value is $f^* \approx -0.965$. The coefficients a_{ij} and c_i are given in Table 5. The tolerance parameter value is $\varepsilon_{12} := 0.1$.

13. *Meyer and Roth problem* (Wolfe 1978) (a least squares problem):

$$\underset{\mathbf{x} \in \mathbb{R}^3}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^5 \left[\frac{x_1 x_3 t_i}{(1 + x_1 t_i + x_2 v_i)} - y_i \right]^2,$$

subject to $x_i \in [-10, 10]$, $i = 1, 2, 3$. The global minimum is located at $(3.13, 15.16, 0.78)^T$ with $f^* \approx 0.4 \times 10^{-4}$. Table 6 lists the coefficient values for this problem. The tolerance parameter value is $\varepsilon_{13} := 0.15$.

14. *Miele and Cantrell problem* (Wolfe 1978):

$$\underset{\mathbf{x} \in \mathbb{R}^4}{\text{minimize}} \quad f(\mathbf{x}) := [\exp(x_1) - x_2]^4 + 100(x_2 - x_3)^6 + [\tan(x_3 - x_4)]^4 + x_1^8,$$

subject to $x_i \in [-1, 1]$, $i = 1, 2, 3, 4$. The number of local minima is unknown but the global minimum is located at $(0, 1, 1, 1)^T$, with $f^* = 0$. The tolerance parameter value is $\varepsilon_{14} := 0.25$.

Table 7 Data for Price's transistor problem

i	g_{i1}	g_{i2}	g_{i3}	g_{i4}
1	0.4850	0.7520	0.8690	0.9820
2	0.3690	1.2540	0.7030	1.4550
3	5.2095	10.0677	22.9274	20.2153
4	23.3037	101.7790	111.4610	191.2670
5	28.5132	111.8467	134.3884	211.4823

15. *Modified Rosenbrock problem* (Price 1977):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := 100(x_2 - x_1^2)^2 + [6.4(x_2 - 0.5)^2 - x_1 - 0.6]^2,$$

subject to $x_i \in [-5, 5]$, $i = 1, 2$. This problem has two global minima located at $(0.3412, 0.1164)^T$ and $(1, 1)^T$, with $f^* = 0$, and one additional local minimum. The tolerance parameter value is $\varepsilon_{15} := 0.005$.

16. *Powell problem* (Rahnamyan et al. 2007):

$$\underset{\mathbf{x} \in \mathbb{R}^{20}}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^{20} |x_i|^{i+1},$$

subject to $x_i \in [-1, 1]$, $i = 1, \dots, 20$. The global minimum is located at $(0, \dots, 0)^T$, with $f^* = 0$. The tolerance parameter value is $\varepsilon_{16} := 0.1$.

17. *Price's transistor problem* (Price 1977):

$$\underset{\mathbf{x} \in \mathbb{R}^9}{\text{minimize}} \quad f(\mathbf{x}) := \gamma^2 + \sum_{k=1}^4 (\alpha_k^2 + \beta_k^2),$$

where

$$\begin{aligned} \alpha_k &= (1 - x_1 x_2) x_3 \left(\exp[x_5 (g_{1k} - 10^{-3} g_{3k} x_7 - 10^{-3} g_{5k} x_8)] - 1 \right) \\ &\quad - g_{5k} + g_{4k} x_2, \\ \beta_k &= (1 - x_1 x_2) x_4 \left(\exp[x_6 (g_{1k} - g_{2k} - 10^{-3} g_{3k} x_7 + 10^{-3} g_{4k} x_9)] - 1 \right) \\ &\quad - g_{5k} x_1 + g_{4k}, \\ \gamma &= x_1 x_3 - x_2 x_4, \end{aligned}$$

and subject to $x_i \in [-10, 10]$, $i = 1, \dots, 9$. The values of the coefficients g_{ik} are given in Table 7. The global minimum is located at $(0.9, 0.45, 1, 2, 8, 8, 5, 1, 2)^T$ with $f^* = 0$. The number of local minima is unknown. The tolerance parameter value is $\varepsilon_{17} := 5$.

18. *Rosenbrock—2-dimensional problem* (Moré et al. 1981) (a special case of problem 16):

$$\underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad f(\mathbf{x}) := [100(x_2 - x_1^2)^2 + (x_1 - 1)^2],$$

subject to $x_i \in [-30, 30]$, $i = 1, 2$. The global minimum is at $(1, 1)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_{18} := 0.01$.

19. *Rosenbrock— n -dimensional problem* (Moré et al. 1981):

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2],$$

subject to $x_i \in [-30, 30]$, $i = 1, \dots, n$ (our tests were performed with $n = 10$). The global minimum is located at $(1, 1, \dots, 1)^T$ with $f^* = 0$; it lies in a narrow, parabolic valley and is therefore difficult to locate. The tolerance parameter value is $\varepsilon_{19} := 1$.

20. *Schwefel problem* (Schwefel 1981):

$$\underset{\mathbf{x} \in \mathbb{R}^{20}}{\text{minimize}} \quad f(\mathbf{x}) := \sum_{i=1}^{20} \left(\sum_{j=1}^i x_j \right)^2$$

subject to $x_i \in [-100, 100]$, $i = 1, \dots, 20$. The global minimum is at $(0, \dots, 0)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_{20} := 0.5$.

21. *Tointgor problem* (Gould et al. 2003): The complete definition of this quadratic minimization problem with 50 variables is given in Toint (1978, Section 3.1). The best solution with $f(\mathbf{x}) = 1175.47$ was obtained by the solver donlp2 (Spellucci 1998). The tolerance parameter value is $\varepsilon_{21} := 5$.

22. *Wood problem* (Wolfe 1978):

$$\underset{\mathbf{x} \in \mathbb{R}^4}{\text{minimize}} \quad f(\mathbf{x}) := 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1),$$

subject to $x_i \in [-10, 10]$, $i = 1, 2, 3, 4$. The global minimum is located at $(1, 1, 1, 1)^T$ with $f^* = 0$. The tolerance parameter value is $\varepsilon_{22} := 0.2$.

References

- Abramson, M. A., Audet, C., Couture, G., Dennis, J. E., Le Digabel, Jr., S., Tribes, C. (2017). *The NOMAD project*. Software available at <https://www.gerad.ca/nomad/>.
- Ali, R., Dhillon, R., El-Gindy, M., Öjjer, F., Johansson, I., & Trivedi, M. (2013). Prediction of rolling resistance and steering characteristics using finite element analysis truck tyre model. *International Journal of Vehicle Systems Modelling and Testing*, 8(2), 179–201.
- Allen, R. G. D. (2008). *Mathematical analysis for economists. Studies in statistics and scientific method* (Vol. 3). London: Read Books.
- Aluffi-Pentini, F., Parisi, V., & Zirilli, F. (1985). Global optimization and stochastic differential equations. *Journal of Optimization Theory and Applications*, 47(1), 1–16.
- Audet, C., & Dennis, J. E., Jr. (2006). Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1), 188–217.
- Benke, K. K., & Skinner, D. R. (1991). A direct search algorithm for global optimization of multivariate functions. *Australian Computer Journal*, 23(3), 73–85.
- Bersini, H., Dorigo, M., Langerman, S., Seront, G., Gambardella, L. (1996). Results of the first international contest on evolutionary optimisation (1st ICEO). In *Proceedings of the IEEE international conference on evolutionary computation (ICEC)*, pp. 611–615. IEEE Press, New York, NY, USA.
- Bessiere, C. (2006). Constraint propagation. In F. Rossi, P. van Beek, & T. Walsh (Eds.), *Handbook of constraint programming*, Ch. 3 (pp. 29–83). Amsterdam: Elsevier.
- Björkman, M., & Holmström, K. (2000). Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering*, 1(4), 373–397.
- Buhmann, M. D. (2003). *Radial basis functions: Theory and implementations*. West Nyack, NY: Cambridge University Press.

- Colson, B., Marcotte, P., & Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research*, 153(1), 235–256.
- Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009). *Introduction to derivative-free optimization, MOS-SIAM series on optimization* (Vol. 8). Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Ding, X.-D., Yao, Z.-G., & Cheng, G. (2009). LINGO language in the logistics distribution center location. *Logistics Engineering and Management*, 10, 335–356.
- Dixon, L., & Szegő, G. (1978). *Towards global optimization* (Vol. 2). New York, NY: North-Holland.
- Dixon, L. C. W., & Price, R. C. (1999). Truncated Newton method for sparse unconstrained optimization using automatic differentiation. *Journal of Optimization Theory and Applications*, 60(2), 261–275.
- Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2), 201–213.
- Dupačová, J., & Popela, P. (2005). Melt control: Charge optimization via stochastic programming. In S. W. Wallace & W. T. Ziemba (Eds.), *Applications of Stochastic Programming* (pp. 277–298). Philadelphia, PA: SIAM.
- Fu, M. C. (2014). *Handbook of simulation optimization. International series in operations research and management science*. New York: Springer.
- Gould, N. I. M., Orban, D., & Toint, Ph L. (2003). CUTer and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4), 373–394.
- Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3), 201–227.
- Holmström, K., Göran, A. (2002). *User's guide for Tomlab v3. 2.11*. <http://tomopt.com/docs/tomlabv32.pdf>. Accessed August 13, 2015.
- Hong, L. J., Nelson, B. L., & Xu, J. (2015). Discrete optimization via simulation. In M. C. Fu (Ed.), *Handbook of simulation optimization* (pp. 9–44). New York, NY: Springer.
- Jakobsson, S., Patriksson, M., Rudholm, J., & Wojciechowski, A. (2010). A method for simulation based optimization using radial basis functions. *Optimization & Engineering*, 4(11), 501–532.
- Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150–194.
- Jansson, C., & Knüppel, O. (1995). A branch and bound algorithm for bound constrained optimization problems without derivatives. *Journal of Global Optimization*, 7(3), 297–331.
- Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 4(21), 345–383.
- Jones, D. R., Perttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1), 157–181.
- Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.
- Kolář, P. (2015). A joint model of heavy truck, tyres, and operating environment for tyres selection. Master's thesis, Department of Applied Mechanics, Chalmers University of Technology, Gothenburg, Sweden.
- Lewis, R. M., Torczon, V., & Trosset, M. W. (2000). Direct search methods: Then and now. *Journal of Computational and Applied Mathematics*, 124(1–2), 191–207.
- Lindroth, P. (2011). *Product configuration from a mathematical optimization perspective*. Ph.D. thesis, Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden.
- Lindroth, P. (2015) TyreOpt—Bränslebesparing med hjälp av däcksennergiförlustoptimering (TyreOpt—Fuel consumption reduction by tyre drag optimization). <http://www.energimyndigheten.se/Forskning/Projektdatabas/>, Project Number: P34882-1, Accessed August 13, 2015 (in Swedish).
- Lophaven, S. N., Nielsen, H. B., Sondergaard, J. (2002). DACE—A MATLAB Kriging Toolbox, version 2.0. Technical report, Department of Informatics and Mathematical Modelling, Technical University of Denmark, Kongens Lyngby, Denmark, August.
- Montaz Ali, M., Khompatraporn, C., & Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31(4), 635–672.
- Moré, J. J., & Wild, S. M. (2009). Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1), 172–191.
- Moré, J. J., Garbow, B. S., & Hillstrome, K. E. (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software (TOMS)*, 7(1), 17–41.
- Powell, M. J. D. (1999). Recent research at Cambridge on radial basis functions. In M. W. Müller, M. D. Buhmann, D. H. Mache, & M. Felten (Eds.), *New developments in approximation theory, ISNM international series of numerical mathematics* (Vol. 132, pp. 215–232). Basel: Birkhäuser.

- Price, W. L. (1977). A controlled random search procedure for global optimization. *The Computer Journal*, 20(4), 367–370.
- Price, W. L. (1983). Global optimization by controlled random search. *Journal of Optimization Theory and Applications*, 40(3), 333–348.
- Rahnamyan, S., Tizhoosh, H. R., & Salama, N. M. M. (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers and Mathematics with Applications*, 53(10), 1605–1614.
- Šabartová, Z., Strömberg, A.-B., Patriksson, M., Lindroth, P. (2014). An optimization model for truck tyres selection. In A. Araujo (Ed.), *Proceedings of the 4th international conference on engineering optimization engineering optimization 2014*, pp. 561–566, CRC Press, Boca Raton, FL, USA. <https://doi.org/10.1201/b17488-101>.
- Schwefel, H.-P. (1981). *Numerical optimization of computer models*. New York, NY: Wiley.
- Shcherbina, O., Neumaier, A., Sam-Haroud, D., Vu, X.-H., Nguyen, T.-V. (2003). Benchmarking global optimization and constraint satisfaction codes. <http://www.mat.univie.ac.at/~neum/glopt/coconut/Benchmark/Benchmark.html>. Accessed August 13, 2015.
- Simpson, T. W., Poplinski, J. D., Koch, P. N., & Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17(2), 129–150.
- Sóbestor, A., Forrester, A. I. J., Toal, D. J. J., Tresidder, E., & Tucker, S. (2014). Engineering design applications of surrogate-assisted optimization techniques. *Optimization and Engineering*, 15(1), 243–265.
- Spellucci, P. (1998). An SQP method for general nonlinear programs using only equality constrained subproblems. *Mathematical Programming*, 82(3), 413–448.
- Stuber, M. D., Scott, J. K., & Barton, P. I. (2015). Convex and concave relaxations of implicit functions. *Optimization Methods and Software*, 30(3), 424–460.
- The MathWorks Inc. (2012). *MATLAB Release 2012b*. Natick, MA: The MathWorks Inc.
- Toint, Ph L. (1978). Some numerical results using a sparse matrix updating formula in unconstrained optimization. *Mathematics of Computation*, 32(143), 839–851.
- Törn, A., & Žilinskas, A. (1989). *Global optimization*. New York, NY: Springer.
- Walter, J. D., & Conant, F. S. (1974). Energy losses in tires. *Tire Science and Technology*, 2(4), 235–260.
- Wendland, H. (2005). *Scattered data approximation*. Cambridge: Cambridge Monographs on Applied and Computational Mathematics.
- Wild, S. M., Regis, R. G., & Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6), 3197–3219.
- Willcox, K., & Wakayama, S. (2003). Simultaneous optimization of a multiple-aircraft family. *Journal of Aircraft*, 40(4), 616–622.
- Wolfe, M. A. (1978). *Numerical methods for unconstrained optimization*. New York, NY: Van Nostrand Reinhold Company.